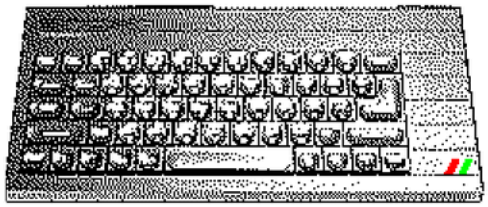
LEC Spectrum R7s

Menü-ROM



Anleitung

Das Boot-ROM des LEC R7s ist auf Platz 8 des 29C040 EEProms gespeichert. Falls am Anschluß für den TEST-Jumper im LEC Spectrum ein Taster angebracht ist, kann dieser Taster beim Einschalten des Spectrums, sowie beim Auslösen eines RESETs gehalten werden, um das BOOT-Rom sofort aufzurufen. Dieser Taster muß solange gehalten werden muß, bis das Menü vollständig auf dem Bildschirm aufgebaut. Bei solch einem „Kaltstart“ ist eine vollständige Initialisierung des Spectrums notwendig.
Das Bootrom verfügt deshalb dann nur über die „Basisfunkionen“.



- Die Wechsel-Taste der ROMPage zwischen 0 und 1 (Taste 8)
- Anwahl der LEC-Speicherverwaltung (Taste L)
- Auswahl des gewünschen ROMs (Tasten 0 – F)
- Dieser Modus wird durch einen blauen Border gekennzeichnet.

Auch der NMI Taster ist in diesem Modus funktionslos.

Im laufenden Betrieb kann das BOOT-Menü durch Eingabe von **OUT 252,8** aufgerufen werden.
Dabei erfolgt kein RESET des Computers.
Die ROMs des LEC R7s sind in 16 (0-15) 32k-Blöcken im 29c040 gespeichert. Sie sind jeweils zwei 16k ROMs auf den Speicherseiten 0 und 1 abgelegt.
Die einzelnen ROMs können durch Eingabe der Tasten 0 – 9 und A – F angewählt werden. Nach Drücken der entsprechenden Taste wird sofort das gewählte ROM angesprungen.
Eine Sonderstellung hat Taste 8. Diese Taste wechselt die ROM-Seiten 0 und 1, siehe Seitenanzeige auf Position 8 rechts.
Da einige ROMs auf diesen Seiten unterschiedlich sind, ist das „Seite 1“ Menü auch anders. Einige im 32k-Modus betriebene ROMs (128k Spectrum, BBC-Basic, SE-Basic) benötigen auf Seite 1 der ROM-Bank ein Sinclair 48k ROM, das wird dann angezeigt und bei der ROM Wahl auch direkt angesprungen.
Bei einem ROM-Wechsel ohne RESET wird die Copyright-Meldung des gewählten ROMs kurz angezeigt.





Die Tasten **L** und **R** sind „Wechsel“-Schalter (EIN = ROT, Aus = Cyan) und beeinflussen das Verhalten des Spectrums beim ROM Wechsel:

R: der ROM-Wechsel erfolgt mit (rot) oder ohne(cyan) RESET des Computers. Vor dem Einsprung ohne RESET in das ROM wird kurz die Copyright-Information des gewählten ROMs eingeblendet.

L: wenn aktiviert (= ROT), wird beim ROMwechsel gleichzeitig der LEC-Speichermodus eingeschaltet, um z.B. CP/M einsetzen zu können.
Achtung.
Die Aufteilung und Nutzung des zur Verfügung stehenden Speichers ist LEC-RAM Modus deutlich anders als im 128 k Modus. Der RAM Bereich von C000h bis FFFFh wird im 128er Modus von Block 0 genutzt. (siehe Speicherschema auf Seite 10) Im LEC-RAM Modus wird an dieser Stelle jedoch Block 1 eingeblendet. Das hat zur Folge, das Software wie MCode Routinen, die im „normalen“ Modus dort gespeichert wurden, nach der Schaltung in den LEC-RAM Modus nicht verfügbar sind.



Um etvl. Abstürze zu vermeiden, wird beim Umschalten in den LEC Modus kurzfristig der RAMTOP auf 49151 herab, und nach dem Moduswechsel wieder auf den alten Wert gesetzt.
(Angezeigt durch die blaue Ecke im Regenbogen-Logo)
Ausserdem sind im LEC-Speichermodus bei allen ROMs nur die Page 1 nutzbar. Durch diese Beschränkung sind einige ROMs, in deren 2. Seite Programme aus dem Z-Menü abgelegt sind, nicht mehr anwählbar und entsprechend gekennzeichnet.

Z - Taste: Blendet eine Auswahlbox mit Zusatzprogrammen ein.
Dieses Z-Menü bietet eine Auswahl von Programmen, die entweder nach entsprechendem Herabsetzen des Ramtop in den oberen Speicherbereich geladen und dann mit verschiedenen ROMs genutzt werden können,
oder andere Hilfs-Programme, die direkt gestartet werden.
Eine Herabsetzung ds RAMTOP wird durch eine blaue Ecke oben im Regenbogenlogo rechts gekennzeichent.



Die mittels Z Menü anwählbaren Zusatzprogramme:

E: erweiterter Microdrive Catalog Befehl und erweiterte Syntax:
S: erweiterte Syntax ohne *c Befehl

Diese beiden Befehle Tasten laden eine Befehlserweiterung in den oberen RAM Bereich.
Die einzelnen Befehle bestehen aus dem * Zeichen, jeweils einem Buchstaben, der sowohl klein wie auch Groß geschrieben werden kann, und evtl. einigen Parametern.
Es sind vor allem Vereinfachungen der Microdrive Befehle und zusätzlich noch einige andere brauchbare Befehlserweiterungen.

Die zusätzlichen Befehle:

*e#	edit Zeile #
*p#,#	16-Bit Poke (# = 16Bit Int)
*P#	double Peek (16Bit Peek)

Die vereinfachten Microdrive-Befehle:
*Bei diesen Befehlen wird das gewünschte Laufwerk als Ziffer vor dem Programnamen innerhalb der „Gänsefüßchen“ angegeben. (Ausnahmen: *r und *c)*

*s"#Name"	speichert "Name" auf Drive #
*l"#Name"	Lädt "Name" von Drive #
*v"#Name"	verifiziert "Name" auf Drive #
*m"#Name“	merged "Name" auf Drive #
*k "#Name"	löscht "Name" (kill) auf Drive # (das „e“ ist schon anderweitig genutzt)
	CODE # / LINE / SCREEN$ wie gewohnt anhängen
*r#	startet das Program "run" auf Drive #. *r ohne # startet „run“ auf Drive 1
*c#	erweiterter Catalog von Drive#



Der Befehl analysiert die Cartridge und zeigt dann Name und Type des Datensatzes.
Basic-Programme werden mit LINE gekennzeichnet, anschließend wird die Autostart-Zeile sowie Größe des Programmes mit und ohne Variablen angegeben.
Bei Maschinencode wird Startadresse und Größe angezeigt, bei anderen Typen (PRINT, DATA) nur die Größe.
In der Kopfzeile wird Drive Nummer und Name der Cartridge angegeben, in der Fußzeile der verbleibende Speicherplatz. Das Programm ist (fast) genau so schnell wie ein normaler CAT Befehl, verbraucht allerdings deutlich mehr Speicher als der Rest der *Befehle. Deshalb kann die erweiterte Syntax auch ohne den *c Befehl geladen werden (**S** anstelle von **E**)
Siehe auch **escat.** Im Vergleich zu **esyntax.** im obigen Screenshot..

R: MICRODRIVE File RENAME

Die Routine ermöglicht ein Umbennenen von Files auf einem Microdrive.
Allerdings benötigt sie einen großen Pufferspeicher, deshalb wird der RAMTOP auf 27999 herabgesetzt.
Diese Syntaxerweterung wird automatisch initialisiert und kann sofort aufgerufen werden:

*r(DriveNr);“alternname“ TO “neuer Name“ (TO als TOKEN). Bsp:
*r1;“test“ TO “TEST“

es gibt allerdings zwei Einschränkungen:
1. DATA Files können nicht umbenannt werden.
2. Die Files müssen kleiner 37kB Größe haben, andernfalls gibt es einen OUT OF MEMORY Fehler.

C: Copy Programm



Das Copy Programm lädt sich in den unteren zwei Drittel des Bildschirmspeichers um das restliche RAM komplett auszunutzen.
Deshalb ist eine Rückkehr in den normalen Betrieb nicht mögich, es muß mittels RESET verlassen werden.
Es ermöglicht das Kopieren zwischen verschiedenen Microdrive Laufwerken und unterstützt auch Kassettenkopien.
Nach Nutzung des Programmes sollte der Rechner neu gestartet werden, um den Speicher wieder frei zu geben.

A: Z80 Assembler EDITAS

Ein Z80 Assembler Programm, das seine Daten auch direkt auf eine Microdrive Cartridge sichern kann.
Eine Beschreibung des Editas würde den Umfang dieser Anleitung sprengen, deshalb gibt es für dieses Programm eine gesonderte Beschreibung.

Z: Z80 Disassembler



Der Disassembler arbeitet komplett im Spectrum Screen.
Der Maschinen-Code liegt im unteren Drittel des Screens.
Dadurch können MC-Programme an allen möglichen Speicherstellen untersucht werden.

ROM-Anwahl

Vor dem Aufruf des Boot-Roms kann mit RANDOMIZE 0-15 ein ROM für den Disassembler gewählt werden, so kann dieses angewählte ROM disassembliert werden. Die Rom-Page 0 oder 1 wird durch das HighByte bei der Randomize -Angabe gewählt. Für die ROM-Page 1 ist deshalb 256 zur Rom Nummer zu addieren.
Randomize 256 lässt den Disassembler somit zum ROM 0 Page 1 wechseln.

Befehle:

Startadresse hexadezimal eingeben und mit ENTER bestätigen.

R: ROM Wechsel (KEIN ROMwechsel des Systems, sondern Angabe der aktuellen ROM-Art!)
Q: Beenden des Disassemblers

V: VDRIVE Treiber nutzen

Der Vdrive Treiber wird auf Adresse 64000 eingeladen, der Ramtop auf 63999 gesetzt.
Übersicht der VDrive Kommandos des VDRIVE Treibers (Quelle: ZX-Heinz)
(die Beschreibung der Befehle gilt ebenfalls für das VDrive ROM (ROM Nr. 5)
Die Grundstruktur der Befehle ist PRINT USR 64E3, STRING.
Die String Kommandos können mit Klein- oder Großbuchstaben eingegeben werden.

Für STRING stehen:
"I" = Initialisierung des Vdrives. Das muss zu Beginn und nach jedem Stick-Wechsel erneut gerufen werden.
Das Vdrive meldet sich mit einigen Angaben zur installierten Firmware und am Ende sollte die Zeichengruppe "D:> erscheinen. Nun ist das VDrive zur Kommandoannahme bereit.
"D" = directory
"R" = read until prompt, liest aus dem Puffer des VDrive bis die Zeichengruppe "D:>" erscheint. VDrive Ready.
"LName" = Lädt das Basic-Programm Name.TAP oder auch ein Codeprogramm Name.TAP.
Der Suffix wird NICHT angegeben.
"LName",XXXX = Lädt ein Code-Programm auf Adresse #XXXX
"SName" = Speichert ein Basic-Programm Name als Name.TAP
"SName",XXXX,YYYY = Speichert ein Code-Programm mit Beginn Adr. #XXXX und Länge YYYY unter dem Namen Name.TAP

***ACHTUNG*: mit diesem Treiber keine Daten auf Adressen über 64000 laden.**
Das überschreibt den Treiber.
Für solche Fälle kann das VDRIVE ROM auf Platz Nr 5 genutzt werden.

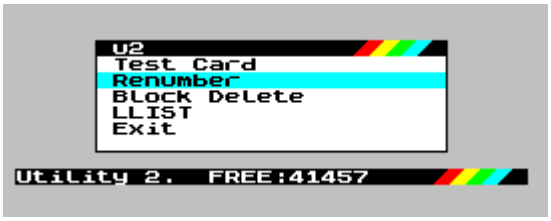
D: Dez/Hex & Hex/Dez Umwandler

Dieses Hilfsprogramm wird im Printerbuffer installiert, der RAMTOP wird nicht verändert.
Deshalb erfolgt die „Ladebetätigung“ mit Angabe der USR Startadressen

```
DezHex:23300   HexDez:23432
```

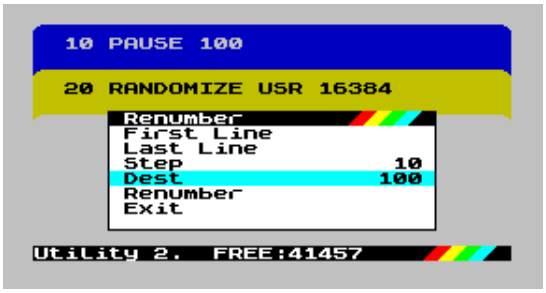
DEC->HEX CONVERTER Start mit Print USR 23300
HEX->DEC CONVERTER Start mit Print USR 23432
Durch Eingabe von Q werden die Routinen beendet

B: RENUMBER (Die BASIC – TOOLs)



Die Basic Tools bestehen im wesentlichen aus einer Renumber und einer Block-Delete Routine. Die Routinen werden über die Pfeiltasten gesteuert, Pfeil auf und ab wechselt die Menüpunkte, Pfeil Rechts – Links steuert verändert die Zahlenargumente im Menüpunkt. Die Schrittweite der Änderung kann mit ENTER zwischen 1 oder 100 (roter Punkt) gewechselt werden. Diese Tools rufen bei EXIT nicht wieder das Boot-Menü auf. Nach dem EXIT befindet man sich im ROM 0 auf Page 0 zum weitereren Editieren oder Starten des veränderten Basic-Programmes.

RENUMBER



Die Routine kann das gesamte Programm oder nur ein Teil davon neu nummerieren.
Das Inkrement zwischen den Zeilen und die Startzeile können angepasst werden.
Alle Verweise auf Zeilennummern, ob intern oder extern zu dem Teil, der neu nummeriert wird, werden geändert, vorausgesetzt, es handelt sich um einfache Ganzzahlen im Bereich von 1 bis 9999, d.h. GO TO, GO SUB, RESTORE, LIST, LLIST und TRUE LINE

Folgendes wird nicht neu nummeriert:
GO TO URL „1000“, GO TO 3E3, GO TO PI * PI, GO TO 70.3, GO TO 99999 sowie
GO TO a*1000.

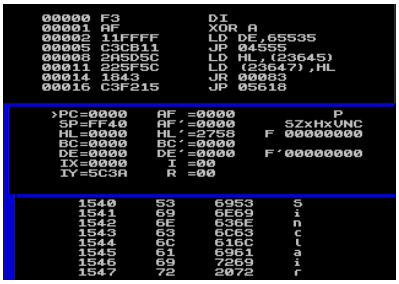
Der Inkrementschritt ist standardmäßig auf 10 eingestellt.
Das Programm kann Zeilen an einen neuen Ort verschieben:Dazu die erste/letzte Zeile festlegen, DANN das Ziel und den Schritt anpassen, dann „Neu nummerieren“ wählen.
Diese Neunummerierung kann im Gegensatz zu einer normalen Umnummerierung mehrere Minuten dauern.
Eine Zeilenverschiebung ohne Schrittweitenänderung erfolgt mit Wahl einer Schrittweite von 0.
Alle Verweise auf Zeilen nach dem Programm werden auf 9999, alle Verweise auf Zeile 0 werden nicht geändert.
Wenn die Neunummerierung oder Verschiebung zu Zeilen größer als 9999 führt oder wenn es zu einem Zeilenkonflikt kommt, wird der Befehl nicht ausgeführt
Falls bei einer Neunummerierung der Speicher voll wird, erscheint die Menüleiste rot. Dann sollte man zu Basic zurückkehren.

BLOCK DELETE



Dieses Modul zeigt die erste und letzte Zeile des Programms an. Aus Sicherheitsgründen kann das gesamte Programm nicht gelöscht werden. Mit den horizontalen Cursortasten, werden die erste und letzte Zeile angepasst.
Dann den grünen Balken auf Löschen bewegen und ENTER drücken. Die beiden angegebenen Zeilen und alle Zeilen dazwischen werden gelöscht. Unten auf der Anzeige wird eine Schätzung des freien Speichers angezeigt. Dieser erhöht sich, wenn Zeilen gelöscht werden, (und kann auch von der Neunummerierung beeinflusst werden.)

F: Der FOX-Monitor



Dieser Monitor/Disassembler arbeitet ie der „X-Disassembler“ direkt im Bildschirmspeicher und kann wie bei diesem beschrieben (ROM-Anwahl), ebenfalls mit unterschiedlichen ROMs genutzt werden. Er besitzt wesenlich mehr Features als der reine Disassembler, benötigt dafür auch die 2 unteren Drittels des Screens für den Code und beschränkt seine Ausgabe deshalb auf das obere Bildschirmdrittel.
Neben dem Disassembling können Regiterinhalte angezeigt und manipuliert werden, der Speicherinhalt als ASCII dargetellt, durchsucht und verändert werden und die Anzeige zwischen Hex und Dezimal gewechselt werden.

Bedienung des FOX - Monitor -

q – Beenden
r - Sprung zu einer Adresse
i - Speicher füllen (Hinzufügen, RL, Löschen usw.)
P – Print
s - Headerless - Block speichern
f - Zeichenfolge suchen (n – nächste)
h - Hex/Dez – Umschaltung der Anzeige
l - Speicherblock verschieben (ldir,lddr)
m - Speicherort eingeben
x - Register untersuchen oder ändern
w - Unterbrechung im RAM setzen
t - Text im RAM eingeben
o – Poke in den Speixher
a - ASCII anzeigen
d - Zwei Zahlen addieren oder subtrahieren
g - Bytes suchen (n - nächste)
j - Headerless - Datei laden
b – Binärkonvertierung
(Tasten 6,7 zum navigieren, i - neuen Wert eingeben, x - zum Monitor zurückkehren)

Alle Optionen können mit EDIT oder SPACE drücken verlassen werden.
Löschen Sie im HEX - Modus #, um eine DEC - Zahl einzugeben. -
Fügen Sie im DEC - Modus # hinzu, um eine HEX - Zahl einzugeben. -
Der Speicher von 22784 bis 65535 steht zur freien Verfügung. -

Der FOX Monitor springt beim Beenden nicht ins Menü zurück, sondern ohne RESET ins aktuellen ROM. Wenn allerdings der Stack oder der Programmcounter verändert wurde, kann er auch im Byte-Nirvana enden.
Eine gewisse Vorsicht beim Manipulieren ist also angebracht!

H: Anzeigen der „Hilfe Seiten“



Mittels der H-Taste können zwei (einhalb) Seiten mit Kurzanleitungen für einige Zusatzroutinen des Menü-Roms angezeigt werden. Bei Anzeige eine Anleitungs-Seite kann mittels N-Taste auf die andere Seite gewechselt werden, mittels Q-Taste wird die Anzeige verlassen und das Menü wieder aufgerufen.

Q: Verlassen des Menüs

Mit dieser Taste kann das Z-Menü ohne Aufruf eines Programmes wieder verlassen werden.

Anmerkung zu den Tools:

Wenn die Routinen in den Arbeitsspeicher kopiert werden, wird dabei wenn nötig, automatisch der RAMTOP geändert. Die Speicherung der Routinen wird mit einer Meldung in der untersten Zeile qutittiert und danach das Z-Menü verlassen.
Diese Meldung wird nicht automatisch gelöscht, verschwindet aber beim Seitenwechsel (Taste 8) oder spätestens beim erneuten Aufruf des „Z-Menüs“.Die Gelbe Markierung rechts unten im Screen zeigt jedoch weiterhin eine in den Speicher geladene Routine an.
Die Meldungen zeigen den USR-Aufruf Adresse der Routine. Der RAMTOP liegt jeweils ein Byte tiefer.

Bitte beachten
Im LEC-RAM Modus wird an im oberen Speicherbereich anstelle Block 0 der Block 1 eingeblendet. Das hat zur Folge, das Software wie MCode Routinen, die im „normalen“ Modus dort gespeichert wurden, nach einer Umschaltung in den LEC-RAM Modus nicht verfügbar sind.

ACHTUNG: Wenn im LEC-Modus Tools genutzt werden sollen, muss vorher der LEC-Modus angewählt werden, dabei wird temporär der Ramtop auf 49151 herabgesetzt!
Dann werden die Routinen für die erweiterte Syntax und der VDrive Treiber in Block 1 geschrieben.
Wenn anschließend der LEC Modus verlassen wird, müssen die gewünschten Tools nochmals geladen werden um verfügbar zu sein.

LEC Monitor (ROM 0 - Page 1, ROM 2 - Page 0/1)

(im LEC ROM mittels gleichzeitigem Drücken von **SYMBOL SHIFT, SPACE** und **B** aufrufbar, über die oben beschriebene NMI-Routine, oder im TOOL ROM mit *m und RAND USR 15101 **)**

Der Monitor verfügt über folgende Befehle (alle Parameter außer Befehl H sind Hex-Zahlen):

B	Exit zu BASIC (falls möglich)
D<hex_num>	Zeigt die eingegebene Hexzahl in Dezimalschreibweise an
G<addr>	Führt ein Unterprogramm an der Adresse G aus. (RET kehrt zum Monitor zurück).
H<dec_num>	Zeigt die eingegebene Dezimalzahl in Hexadezimalschreibweise an
M<addr>	Hex- und ASCII-Dump von der angegebenen Adresse (SPACE für Exit).
R	Register AF, BC, DE, HL, IX und SP anzeigen und/oder ändern.
S<addr>	Patcht den Speicher von der angegebenen Adresse mit dem Hex-Editor (einfach eingeben). SPACE beendet den Hex-Editors).

Nutzung der Schriftgrößen im 64ZpZ -ROM (ROM 7 - ROM-Page 0):

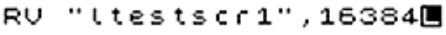
Das 64/32 Zeichen pro Zeile ROM startet im 64 Zeichen/Zeile Modus.
In diesem ROM ist der COPY Befehl ersetzt durch einen Befehl zum Moduswechsel. Ein Druck auf die Z-Taste (Anzeige wZpZ = wechsel Zeichen pro Zeile) gefolgt von ENTER wechselt in den 32 Zeichen/Zeile Modus und auch wieder zurück.



Zur Erinnerung wird das in der Copyright Meldung gezeigt.
Ausserdem kann mittels POKE 23611,220 in den 64 ZpZ- Modus und mit POKE 23611,204 in den 32 ZpZ- Modus geschaltet werden, falls in einem Programm ein bestimmter Modus erzwungen werden soll.

VDRV Befehl im VDrive2 ROM (ROM 5 - ROM-Page 0/1):

In diesem ROM ist der COPY Befehl ersetzt durch einen VDRV Befehle.
Anstelle von PRINT USR 15e3,"LTESTSCR1",16384 zum Laden eines Testbildes (hier als Befehlsbeispiel) kann jetzt VDRV "ltestscr1",16384 eingegeben werden.
Die Eingabe von PRINT USR 15e3 wird ersetzt durch einen Druck auf die Z-Taste.
Kleinbuchstaben können ebenfalls verwendet werden.



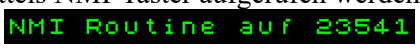
Wer sich nicht umgewöhnen möchte, kann natürlich die alte Syntax weiterhin verwenden!

Move Befehl des LEC-ROM V1.6 (ROM 2 - ROM-Page 0/1):

MOVE „<src_code><src_addr>,<src_len>“, „<dest_code><dest_addr>“
Dadurch werden <src_len> Bytes von <src_addr> nach <dest_addr> verschoben.
<src_code> und <dest_code> : „r“ = „Standard-ROM/RAM“ , „d“ = „zusätzlicher LEC RAM“.
Beispiel: MOVE „r5010,424“, „r16384“
Zwischen Anführungszeichen sind Ausdrücke erlaubt. Andere MOVE-Formen funktionieren wie gewohnt.
Wenn nicht das zusätzliche LEC-RAM angesprochen wird, funktioniert der Befehl auch ohne eingeschaltete LEC-Speicher Verwaltung.

Funktion des NMI-Tasters im BOOT-ROM

Im Gegensatz zu früheren Bootrom Versionen ruft ein Drücken des NMI-Tasters nicht direkt den Monitor im LEC-Rom auf, sondern legt eine Routine zum Aufruf dieses Monitors in das Ende des Printerbuffers (11 Bytes ab 23541) und ändert die Systemvariable NMIREG auf diese Adresse. Das hat den Vorteil, das diese Routine bei einem ROM-Wechsel ohne RESET erhalten bleibt, und auch von anderen ROMs mit korrekt funktionierender NMI-Routine (z.B. alle SINCLAIR ROMs) mittels NMI-Taster aufgerufen werden kann.
Das Einladen der Routine wird in der untersten Zeile angezeigt,



und das automatische RESET beim ROM-Wechsel ausgeschaltet. Das Menü-ROM wird nicht verlassen.

Das Sinclair 48K Rom mit Zusatz Routinen (ROM 0 - ROM-Page 1)

Dies ROM ist bei der Entwicklung des Bootrom entstanden und stellt einige bei der *Syntax erklärte Befehle zur Verfügung. Diese werden jedoch nicht im Oberen RAM-Bereich, sondern im Printerbuffer gespeichert.

Der Monitor kann auch über den NMI Taster aufgerufen werden. Der NMI ist in diesem ROM „hardcodiert“. Das bedeutet, bei diesem ROM wird durch einen ausgelösten NMI **immer** der Monitor aufgerufen, egal was in der Systemvariable NMIREG gespeichert ist, in diesem ROM wird diese Systemvariable nicht beachtet. Das ermöglicht den Monitoraufruf auch bei korumpierten Systemvariablen.

Die Adresse zum Wiederaufruf des EDITAS Assemblers kann genutzt werden, um den EDITAS wieder aufzurufen, wenn er mittels RETURN verlassen wurde.



Wird dieses ROM mit leerem Printerbuffer gestartet, so werden die zusätzlichen Befehle sofort initialisiert und eine kleine Hilfestellung wird am Screen gezeigt.
Befinden sich schon andere Routinen im Printerbuffer, so werden diese nicht überschrieben.

In diesem Fall können die Erweiterungen mit RAND USR 15e3 geladen und initialisiert werden.
In der Copyrightzeile wird darauf auch hingewiesen.
Der LEC-Monitor kann ebenso mittels RAND USR `©Sinclair USR 15000/101 *Tool` 15101 und mit *m gestartet werden..
Ein Vector auf eine evtl. im oberen RAM befindliche Routine (z.B. Ext. CAT-Syntax) wird gesichert und kann mit *w wieder gesetzt werden.

Einige Hinweise:

Betrieb mit deaktiviertem Interface 1:



Da einige Erweiterungen für den Expansions-Port des ZX Spectrum, z.B. das DIVMMC, nicht mit dem Interface 1 kompatibel sind, besteht die durch eine Modifikation des Systems die Möglichkeit, das Interface 1 zu deaktivieren.
Eine Testroutine im Menü-ROM erkennt dieses und blendet daraufhin eine Meldung in der untersten Bildschirmzeile ein. Ausserdem sind alle Tools, die ein Interface 1 zum Betrieb brauchen, dann im Z-Menü ausgeblendet und nicht wählbar.
Das Z-Menü ist dementsprechend reduziert.

Ein Seiteneffekt der LEC- Speicherverwaltung im R7s:

Einen Seiteneffekt vom LEC R7s III wird bei einigen ROM sichtbar:
Wenn mittels Menü, oder mittels OUT 252,128 der LEC Speicher-Modus angewählt wird, schaltet der LEC R7s gleichzeitig auf die ROM-Page 1.
Das bedeutet, bei Einsatz der LEC-Speicherverwaltung sind nur die ROMs auf Page 1 nutzbar!
Bei den normal gedoppelten ROMs fällt das nicht auf, aber bei ROMs mit unterschiedlicher Belegung auf Page 0 und 1.

Reset im Diagnostic ROM

Das Retroleum Diagnostic ROM ist nicht originär für einen Einsatz in einem Gerät vit vielen unterschiedlichen ROMs entwickelt worden, sondern primär zum Nutzen mit der Smart-Card von Retroleum.
Der RESET im Menü des ROMs bewirkt deshalb **nicht** einen RESET des LEC-Spectrums, sondern lediglich einen RESET des ROMs an sich = Neustart der Diagnose.
Das Retroleum Diagnose ROM kann nur mittels RESET-Taster oder durch Ausschalten des Gerätes verlassen werden.

Zur Nutzung der Systemvariable SEED

Die SysVariable SEED (23760 = 5c76h) kann durch RANDOMIZE „Zahl“ belegt werden, und wird von den Routinen im Bootrom genutzt. (X-Disassembler, FOX-Monitor,).
Sie kann auch genutzt werden, um eine 16bit Zahl in ihre High- und Low-Byte Komponenten aufzuspalten.
RANDOMIZE X: PRINT PEEK 23670, PEEK 23771 gibt das LowByte und das Highbyte von X aus.
RANDOMIZE X: POKE adr1,PEEK 23670: POKE adr2,PEEK 23771 ermöglicht es, ohne große Umrechnung einen 16bit Wert in den Speicher zu poken.
ACHTUNG: RANDOMIZE 0 setzt SEED nicht auf 0 sondern belegt es mit einer Zufallszahl.
Zum Löschen müssen die Speicherstellen 23670 und 23671 explizit mit 0 gepoked werden.

Die Belegung des 29C040

Adresse dezimal	Adresse hex	ROMNr Steuerung OUT 252, 0 -15	Menü Taste	Seite Steuerung OUT 253, 0 od. 16 (Bit 4)	ROM
0	0	0	0	0	Sinclair 48k (original)
16384	4000		0	1	Sinclair 48K (NMI-Monitor und Tools)
32768	8000	1	1	0	ZX Spectrum 128 Rom 0 (Editor)
49152	C000		1	1	ZX Spectrum 128 Rom 1 (48k Basic)
65536	10000	2	2	0	LEC V1.6 ROM
81920	14000		2	1	LEC V1.6 ROM
98304	18000	3	3	0	ZX48 V2 (R.Guerra) ROM
114688	1C000		3	1	ZX48 V2 (R.Guerra) ROM
131072	20000	4	4	0	Gosh wonderfull ROM V1.33
147456	24000		4	1	Gosh wonderfull ROM V1.33
163840	28000	5	5	0	VDrive2 15VDR2 ROM
180224	2C000		5	1	VDrive2 15VDR2 ROM
196608	30000	6	6	0	XUP03 ROM
212992	34000		6	1	XUP03 ROM
229376	38000	7	7	0	64 Zeichen/Zeile ROM
245760	3C000		7	1	GAMA_81 ROM
262144	40000	8	8	0	Menü ROM Page 0
278528	44000		8	1	Menü ROM Page 1
294912	*48000*	*9*	*9*	*0*	IMC (Ian Collier) ROM
311296	4C000		9	1	IMC (Ian Collier) ROM
327680	50000	10	A	0	Looking Glass ROM
344064	*54000*			*1*	*ROM für Zusatzroutinen (nicht wählbar)*
360448	58000	11	B	0	JGH V0.76 ROM
376832	*5C000*			*1*	*ROM für das EDITAS Prg (nicht wählbar)*
393216	60000	12	C	0	OpenSE-Basic ROM
409600	64000		C	1	Pretty Basic ROM
425984	68000	13	D	0	BBC-Basic ROM
442368	6C000		D	1	Sinclair 48k (original)
458752	70000	14	E	0	Magician Game
475136	74000			1	*Kurzanleitungen (nicht direkt wählbar)*
491520	78000	15	F	0	Retroleum Diagnostic
507904	7C000		F	1	Retroleum Diagnostic
524287	*7FFFF*				*(ENDE)*

Für den Betrieb des LEC-Menüs sind neben den ROMs auf Platz 8, die beiden Sinclair ROMs auf Platz 0, sowie die ROMs auf Seite 1 von Platz 10 (=A), Platz 11 (=B) und Platz 14 (=E) notwendig. Alle anderen ROMs sind beliebig austauschbar.
Sinnvollerweise sollte dann auch die Menübeschriftung angepasst werden.

Der Speicher des LEC R7s und seine Ansteuerung mittels OUT 252 (FCh) und OUT 253 (FDh)

Der LEC R7s enthält zwei völlig unterschiedliche Arten, den zusätzlichen Speicher anzusprechen.

Der erste Modus ist die vom Spectrum 128k bekannte Speicherverwaltung.

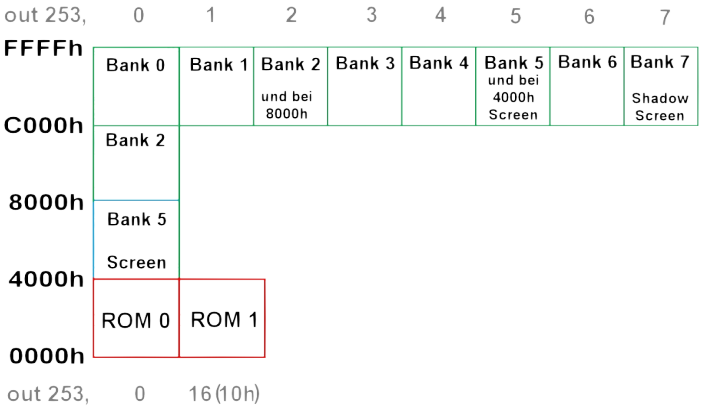
In diesem Modus startet der LEC R7s nach dem Einschalten.

Der „128er"-Speichermodus ist im LEC-R7s allerdings nicht auf den Betrieb mit dem 128k ROM beschränkt sondern steht auch allen 48K ROMs zur Verfügung. Smit sind alle Programme, die nicht auf das 128k Sinclair Basic angewiesen sind, auch mit 48k ROMs nutzbar.

Der zusätzliche Speicher wird in 16kB großen Blöcken in den oberen Adressraum von C000h – FFFFh = 49152 – 65535 eingeblendet. Dabei ist zu beachten, das Bank2 gleichzeitig im Adressraum 8000h-BFFFh = 32768 – 49151 und Bank 5 gleichzeitig im Adressraum 4000h – 7FFFh = 16384 – 32767 (und damit auch im Bildschirmspeicher) eingeblendet sind.

(Display 16384-22391, Attribute 22528 – 23295, auch auf 49152- 55159 und 55160-56063 in Bank 5)

Der LEC-R7s Speicher im Spectrum 128 - Modus



Beim Auslagern des Speichers sollten Interrupts deaktiviert werden und der Stack sollte sich in einem Bereich befinden, der sich nicht ändert.

Wenn normaler Interruptcode ausgeführt werden soll, muss die Systemvariable bei **5B5Ch (23388)** mit dem letzten an Port 7FFDh gesendeten Wert aktualisiert werden. Dieser Port kann nicht gelesen werden. Auf 128 und +2 wird der Speicher vollständig von Port 7FFDh gesteuert. Das an **Port 7FFDh (32765** – es funktioniert genauso gut mit **OUT 253,x)** auszugebende Byte wird folgendermaßen interpretiert:

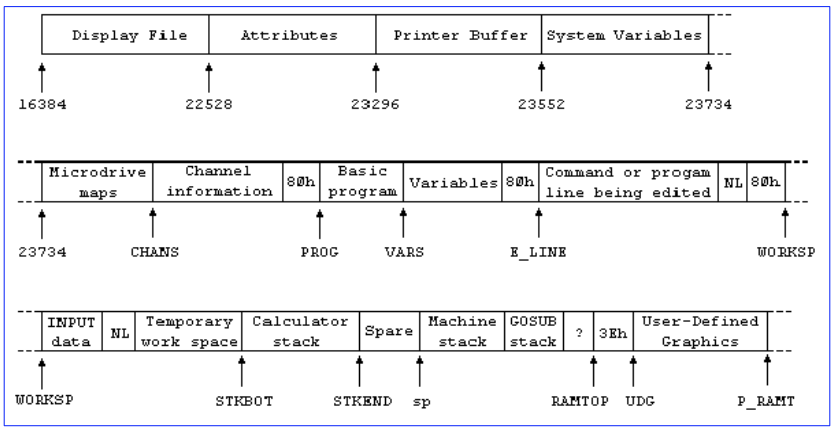
Bits 0-2 RAM-Seite (0-7), die in den Speicher bei 0C000h abgebildet werden soll

Bit 3 Auswahl des anzuzeigenden normalen (0) oder Schattenbildschirms (1). Der normale Bildschirm befindet sich in Bank 5, während der Schattenbildschirm in Bank 7 ist. Der Speicher zwischen 0x4000 und 0x7fff wird nicht nicht beeinflusst, (immer Bank 5).

Bit 4 Auswahl der ROM-Seiten 0 oder 1

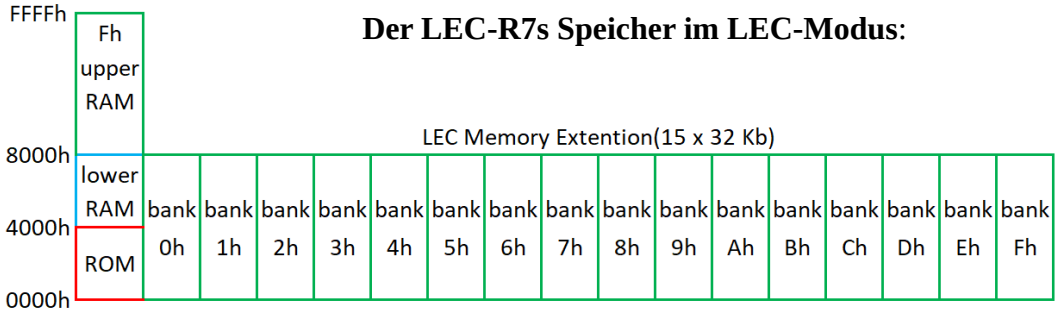
Bit 5: ist im LEC- R7s außer Funktion gesetzt (Sperrt im 128er Spectrum das Bankswitching im 48k-Modus)

Im 128er Modus wird der Speicher wie im Bild angegeben benutzt. Im LEC-RAM Modus ist dagegen der Speicher von Bank 1 in den Bereich von C000h bis FFFh eingeblendet. Da sich unterhalb des RamTop vom System des Spectrums genutzte Bereiche befinden (Gosub-Stack, ERRSP…) kann der Spectrum abstürzen, wenn im Betrieb der Speichermodus mit OUT 252,[Bit 7] gewechselt wird und der RamTop oberhalb BFFFh positioniert ist. Zum Modus-Wechsel, wie auch zum BankSwitching im 128er Modus sollte deshalb der RamTop auf maximal 49151 (=BFFFh) gesetzt sein, um einen Systemcrash zu vermeiden.



Der zweite Speichermodus ist die schon vom LEC R3 bekannte Speicherorganisation der LEC-Speichererweiterung. In diesem Modus arbeitet der Computer permanent mit den ROM-Pages 1.
In diesem Modus kann der Zusatzspeicher in 32k Blöcken in den untereren Adressraum des Spectrum (0000h – 7FFFh = 0 – 32767) eingeblendet werden.
Damit sind dann sowohl ROM als auch die unteren 16k des normalen RAM (Bildspeicher, Druckerpuffer, Systemvariablen, Channel-Informationen und unterer Basic-Bereich) ausgeblendet. Dieser „ALL RAM"-Modus kann sinnvoll nur von Maschinencode Routinen im Speicherbereich ab 8000h (32768) oder von einem Betriebssystem innerhalb des LEC-Speichers (z.B. CP/M 2.2) genutzt werden.

Der LEC-R7s Speicher im LEC-Modus:



out 253,	0h	80h	88h	90h	98h	A0h	Abh	B0h	B8h	C0h	C8h	D0h	D8h	E0h	E8h	F0h	F8h
out 253,	0	128	136	144	152	160	168	176	184	192	200	208	216	224	232	240	248

Speicheraufteilung und Addressierung der Speicherbänke: im LEC (=AllRam) Modus sind Lower Ram und ROM ausgeblendet und eine RAM Bank der Bänke 0 -15 in diesem Speicherbereich eingeblendet
ACHTUNG: *die letzte RAM Bank (15) ist permanent auf die Adressen 8000h - FFFFh eingeblendet, eine Wahl der RAM-Bank 15 bewirkt deshalb, das diese Bank zwei mal im RAM Bereich eingeblendet ist!*

Steuerung über OUT 253,x und Out 252,x

Out 252, (Bit7) Wechselweise umschalten LEC-Speicher Modus / 128er Modus
Bit 7=1 Freigabe für LEC-RAM Modus, Bit 7=0 Freigabe 128er RAM Modus
Nur im 128er RAM-Modus ist ebenfalls der AY-Soundport nutzbar

Out 252, (Bit 0-3) Wahl des aktuellen ROMs (0 – 15)

Out 253, (Bit7) All-RAM Modus (Bit 7=1 : ein / Bit 7=0 : aus)

Out 253, (Bit 3 - 6) Wahl der RAM Seite 0 – 15
Ein Einschalten des All-Ram Modus mit OUT 253,128 wählt also gleichzeitig Bank 0 des LEC-Spechers an. Aus Basic heraus macht dieser Befehl keinen Sinn, da man sich damit den Basic-Interpreter wegschaltet und das System einfriert!

Beispiele: *(binär Bits* `76543210`*)*

Out 253,128 (binär `10000000`) AllRAM – Ram Bank 0 | Out 253,240 (binär `11110000`) AllRAM – Ram Bank 14
Out 253,136 (binär `10001000`) AllRAM – Ram Bank 1 | Out 253,0 (binär `00000000`) AllRAM aus

Out 252,128 (binär `10000000`) LEC Modus – ROM 0 | Out 252,15 (binär `00001111`) 128er Modus – ROM 15
Out 252,140 (binär `10001100`) LEC Modus – ROM 12 | Out 252,4 (binär `00000100`) 128er Modus – ROM 4

Speicheradressen und Ansteuerung der ROMs im W29C040 EEProm

	Seite 0	Seite 1	ROM-wahl
ROM 0	0	16384	out 252,0
ROM 1	32768	49152	out 252,1
ROM 2	65536	81920	out 252,2
ROM 3	98304	114688	out 252,3
ROM 4	131072	147456	out 252,4
ROM 5	163840	180224	out 252,5
ROM 6	196608	212992	out 252,6
ROM 7	229376	245760	out 252,7
ROM 8	262144	278528	out 252,8
ROM 9	294912	311296	out 252,9
ROM 10	327680	344064	out 252,10
ROM 11	360448	376832	out 252,11
ROM 12	393216	409600	out 252,12
ROM 13	425984	442368	out 252,13
ROM 14	458752	475136	out 252,14
ROM 15	491520	507904	out 252,15
Seiten-wahl .	Out 253,0 Bit 4 = 0	Out 253,16 Bit 4 = 1	

Die Grafik zeigt den Beginn der ROM Speicheradressen (dezimal) im 29c040 Chip. (z.B. beim Auslesen oder Programieren des Chips)
Die einzelnen 32 K Bänke werden über die Bits 0-3 bei Port 252 (OUT 252,0-15) angewählt.
Die einzelnen 16 K Seiten werden über Bit 4 bei Port 253 gewählt (OUT 253,0 oder 16)

Das gewählte ROM belegt wird dadurch auf den Adressbereich 0 – 16384 (0 – 4000h) im Spectrum eingeblendet.
Das entspricht dem roten ROM bzw. ROM 0 der Zeichnungen auf Seite 8 und 9.

Beim Einschalten und nach einem RESET des LEC-R7s-Spectrum ist immer ROM 0 mit Seite 0 in diesem Adressbereich eingeblendet.

Zum Beachten:
Der Port 253 wird mit den Bits 0-3 auch anderweitig benutzt. (Siehe Seite 9)

Manipulation des Bootmenüs mittels Hex-Editor und Spectrum Emulator

Falls jemand die ROMs in der Sammlung austauscht, kann der Menüscreen wie folgt geändert werden:

1. beide Menüroms mittels eines Hexeditors extrahieren, (z.B. HxD),
2. aus den Menüroms die Menüscrens (2000h, 1b00h Länge) extrahieren und diese mittels eines Emulators manipulieren. Mit dem FUSE Emulator. damit geht das wie folgt:
   ein Basicprogram eingeben:

   10 PAUSE 100
   20 PRINT PAPER 0; INK (gewünschte Farbe); AT (gewünschte, Position);"gewünschter Text"
   30 goto 30

   Das Programm mit run starten und innerhalb der Pause den Screen (File - Load binary data...) auf die Adresse 16384 (Beginn Bildschirmspeicher) laden.
   Nach der Pause wird dann der gewünschte Text in den geladenen Menüscreen geschrieben.
   Durch die Endlosschleife in Zeile 30 kann man jetzt schauen, ob der gewünschte Text an der richtigen Stelle steht.
3. Falls ja, den Screen mit File - Save binary data... (Start 16384, Länge 6912) wieder speichern.
4. Das Bootrom im HexEditor wieder zusammenbasteln (mit den gesichertem Binärfile ab Adresse 2000h überschreiben)

ACHTUNG: die „Leeren“ Zeilen im Menü sind mit Code belegt und die beiden ersten Bytes im Screen (Links Oben = oberste Zeile Position 0 und 1 werden durch Flags benutzt.
Also möglichst nur die vorhandenen Textzeilen ändern.
Das folgende Bild des Menüs ohne Attribute zeigt die Code-Belegung innerhalb des Screens:

